**Journal of Education and Educational Research**
ISSN: 2957-9465 | Vol. 18, No. 2, 2026

# Project-based Teaching Reform of Java Course: Knowledge Graph Empowering Flipped Classroom 2.0

**Cui Dong, Wenxi Mo**

Guangdong University of Science and Technology, Dongguan, Guangdong, 523000, China

**Abstract:** At present, the teaching of Java courses is mainly based on the "lecture-based" model. It not only faces the challenges of abstract theoretical knowledge and students' difficulty in transforming grammar points into practical engineering capabilities, but also fails to meet the demand for cultivating students' innovative ability. This paper introduces a teaching practice model that integrates the concept of flipped classroom and the project introduction method. We select the "Student Achievement Management System", which is close to the actual school life, as the practical carrier throughout the entire course, to realize the three-in-one integrated teaching of project introduction (PBL), knowledge graph construction, and flipped classroom deepening, helping students achieve the transformation from code writers to system builders.

**Keywords:** Java Courses; PBL; Knowledge Graph; Flipped Classroom.

## 1. Introduction

Against the backdrop of the in-depth integration of the digital economy and intelligent manufacturing, Java programming language, as one of the core technologies in the field of software development, the teaching quality of its corresponding courses is directly related to the cultivation effect of engineering practice capabilities and innovative thinking of students majoring in computer science and related disciplines, and serves as a key link connecting professional foundations and industrial demands. Currently, the course "Java Programming" in most domestic universities still generally adopts the traditional lecture-based teaching mode, which focuses on teachers' one-way knowledge output. Although this mode can ensure the systematicness of knowledge transmission, it is difficult to adapt to the essential requirement of "high integration of theory and practice" in programming language teaching, and has gradually exposed many prominent problems. On the one hand, the core theoretical knowledge of Java courses, such as object-oriented thinking, exception handling, and multi-threading, is highly abstract. In lecture-based teaching, students are mostly in a passive receiving state, making it difficult to establish the connection between grammatical rules and practical application scenarios. This leads to the common dilemma of "understanding grammar but not being able to program, or being able to program but not understanding engineering", and failing to effectively convert scattered grammatical points into practical engineering capabilities to solve complex problems[1]. On the other hand, the traditional mode overemphasizes knowledge indoctrination, neglects the cultivation of students' independent inquiry and innovative thinking, and thus cannot meet the training requirements for high-quality applied and innovative talents in the new era, which deviates from the guiding principle of the Ministry of Education of "deepening the integration of industry and education and promoting the reform of education methods".

To address the above teaching dilemmas, this paper proposes a teaching practice mode integrating the flipped classroom concept and project introduction method from the perspective of the actual teaching situation of the "Java Programming"[2] course, aiming to solve the traditional teaching problems through the "trinity" integrated design. Considering the applicability and relevance of the teaching carrier, the "Student Performance Management System", which is closely related to students' campus life, is selected as the core practical carrier throughout the course. This carrier covers typical functional modules such as user login, score entry, query and statistics, and data maintenance, and can fully cover the core knowledge points and technical application scenarios of the Java course. Through flipped classroom, the teaching process of "pre-class, in-class and after-class" is reconstructed, transferring the knowledge preview link to online independent completion, and focusing classroom time on interactive discussion and problem-solving, which effectively highlights the dominant position of students. Taking the real project "Student Performance Management System" as the core carrier, it guides students to integrate knowledge and temper their abilities in the process of completing project tasks, realizing the organic unity of knowledge learning and engineering practice. At the same time, the visualization function of knowledge graph is used to present knowledge connections, helping students build cognitive scaffolding and lower the threshold for understanding abstract concepts. Students can achieve meaningful knowledge construction through independent exploration of the graph (e.g., jumping from interface nodes to abstract class comparison). Through systematic teaching practice design, this paper explores effective paths to improve the teaching quality of Java courses and helps students realize the ability leap from "code writers passively accepting knowledge" to "system builders actively solving problems".

## 2. Teaching Reform Scheme

In this paper, we use knowledge graph to construct the overall course knowledge and project analysis knowledge, take "Flipped Classroom 2.0" as the theoretical framework, and combine it with Project-Based Learning (PBL)[3], to construct a full-process blended teaching model of "Pre-class - In-class - Post-class". We realize the new teaching concept of "student-centered, project-driven, and ability-oriented".

## 2.1. Knowledge Graph Guided Project: Constructing Systematic Cognition of Java Technology Architecture

The Java technology stack has complex and diverse content, ranging from basic syntax to enterprise-level architecture, involving a large number of knowledge points with strong correlations. In the learning process, students often fall into the dilemma of mastering a single technical point in isolation, but failing to understand its position and role in the overall architecture, and they are even at a loss when facing actual project development. In the course practice session, some students, when facing specific project requirements, find it difficult to connect scattered knowledge points to form solutions. As a structured knowledge representation method, knowledge graph can connect scattered Java technology knowledge points with practical applications, closely link them with the actual requirements of the project, and form a cognitive body with clear levels and coherent logic. This paper selects the library management system, which is close to students' campus life, as the overall teaching project.

In this paper, we take knowledge graph[4] as the guide, combine the entire development process of the library management system project, and adopt the Java technology architecture knowledge graph of this project. Its core value lies not only in "connection" and "layering", but also in "demand orientation" .It makes technical learning always focus on the actual requirements of the project, avoiding abstract learning that is divorced from practice. The connection includes breaking the isolated state between knowledge points, enabling students to understand "why learn this", "what problems can this technology solve", and "how it cooperates with other technologies". Starting from the "bottom foundation layer" of the knowledge graph, we sort out the core foundation of the Java technology architecture, combined with the basic functional requirements of the library management system project. This layer is the cornerstone of the entire Java technology system, and all upper-level technologies are built on it. Java basic syntax is the "stepping stone" for entry. Students need to master basic operations such as variables, data types, and process control. For example, when implementing the reader information addition function, we need to use string type to store reader name and reader ID, and integer type to store reader category ID and borrowable quantity, and use conditional judgment statements to verify the validity of the input information (such as whether the reader ID conforms to the school's coding rules, and whether the borrowable quantity is within a reasonable range). When implementing the book information addition function, we can also use string to store the book ISBN, book title, and author information, and use floating-point type to store the book price, use conditional judgment to verify the format validity of ISBN, and use loop statements to realize the batch entry of book information, so that students can understand the practical role of basic syntax in the implementation of specific functions.

Object-oriented thinking is the core design concept of the Java language, and also the core method for constructing the entity model of the library management project. During the course teaching, instructors abstract business entities such as readers, books, borrowing records, and system administrators in the real world into corresponding classes, so that students can better understand. Each class encapsulates corresponding attributes and methods. For example, the Reader class encapsulates attributes such as reader ID, name, affiliated department, and registration time, as well as methods such as "query personal borrowing records" and "handle borrowing renewal"; the Book class encapsulates attributes such as book ISBN, title, author, category ID, stock quantity, and status mark, as well as methods such as "update stock status" and "mark borrowing / in-library status"; the Borrowing Record class encapsulates attributes such as borrowing ID, associated reader ID, associated book ID, borrowing time, and due time, as well as methods such as "calculate overdue days" and "calculate overdue fine"; the System User class encapsulates attributes such as user account, encrypted password, and permission role, as well as methods such as "modify personal password" and "exit the system". Through the three major characteristics of encapsulation, inheritance, and polymorphism, we can not only improve the reusability and maintainability of the code, but also make the code structure more consistent with the business logic of the project. This is also the foundation for understanding the framework design idea in the follow-up learning. The example of the basic syntax knowledge graph for project splitting is shown in the following figure:

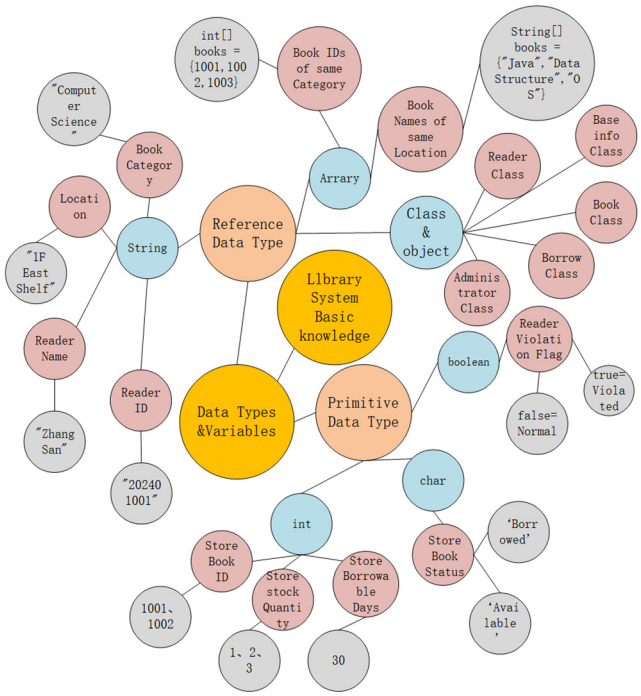



**Fig 1.** Basic Knowledge Graph Structure Diagram of Library Management System

## 2.2. Flipped Classroom 2.0 Model: Online Independent Foundation Building and Offline In-depth Interaction Page

Aiming at the teaching pain points of the "Java Programming" course, such as scattered knowledge points and the disconnection between practice and theory, this paper constructs the Flipped Classroom 2.0 teaching model [5] of "Online Independent Foundation Building + Offline In-depth Interaction". It moves the knowledge transfer link of the traditional classroom to the front, releases classroom time for in-depth practice and discussion, and realizes the reconstruction and upgrading of the teaching process.

### 2.2.1. Online Independent Foundation Building

Fragmented preview with micro-videos: We disassemble the core knowledge points of the Java course into lightweight

micro-videos of 5-8 minutes, covering from basic syntax such as "variables and data types" and "conditional judgment statements" to advanced development technologies such as "transaction management" and "dynamic SQL". Students can arrange their learning time independently according to their own learning rhythm, flexibly complete the preview of basic content, and adapt to the learning needs of students with different foundations.

Knowledge graph guided learning: With the supporting Java technology knowledge graph of the library management system, when students preview a single knowledge point, they can intuitively see the position and correlation of this knowledge point in the project through the graph. For example, when learning "conditional judgment", they can simultaneously see that this knowledge point can be used in practical scenarios such as parameter verification for reader information addition, and business rule verification for book borrowing; when learning "transaction management", they can clearly see that this technology is the core support to ensure the data consistency of book borrowing and returning services. It breaks the isolated state of knowledge points, allows students to establish the complete correlation cognition of "knowledge point - function - project" in advance[6], and avoids abstract learning divorced from practice.

Interactive preview on online platform: Relying on the online learning website, we provide students with content such as project requirement documents, function structure diagrams, and preview self-test exercises. Students can familiarize themselves with the five core modules and functional requirements of the library management system in advance, complete the preview exercises to test their own learning effects, and at the same time, they can mark the questions encountered during the preview process on the platform, and enter the offline classroom with clear questions.

#### 2.2.2. Offline In-depth Interaction: Project-oriented Integrated Teaching to Realize Ability Transition

Offline classrooms no longer carry out repeated teaching of basic knowledge points, but focus on the in-depth integration of projects and course knowledge, carry out high-level discussion and practical training, and fully highlight the dominant position of students:

1. Problem Discussion and Knowledge Internalization: At the initial stage of the class, aiming at the common questions marked by students before class, we carry out group discussion and collective Q&A, help students clear the knowledge obstacles of preview, complete the internalization and absorption of basic knowledge points, and prepare for the follow-up practical training session.

2. Brainstorming for High Concurrency Scenarios: Focusing on the real business pain points of the library management system, we carry out brainstorming for high concurrency solutions. For example, for the high concurrency scenario of "freshmen concentrating on borrowing at the beginning of the semester and students concentrating on returning books at the end of the semester", we guide students to combine the knowledge they have learned to discuss how to solve concurrency problems through lock mechanism, cache optimization, transaction isolation level adjustment, interface current limiting and other solutions, so that students can jump out of basic code writing, think about problems from the perspective of system architecture, and exercise their ability of problem analysis and scheme design

3. Grouped Code Practice for Projects: We divide students into groups, and carry out code practice around the modules of the library management system. Each group is responsible for implementing the functions of a core module: some groups are responsible for the reader information management module, and implement the CRUD functions of reader information with the knowledge points learned before class, such as basic syntax and entity encapsulation; some groups are responsible for the book borrowing module, implement knowledge points such as transaction management, date API, and fine calculation, and realize the complete business process of borrowing and returning; some groups are responsible for the user management module, implement knowledge points such as password encryption and session management, and realize the user permission management function.

In the process, instructors carry out targeted guidance, guide students to integrate scattered knowledge points into complete function implementation, help students complete the transformation from theory to practice, and finally help students realize the ability transition from "code writers who passively accept knowledge" to "system builders who actively solve problems"

### 2.3. Project Introduction: Preparing for Comprehensive Project Development

Combined with the function structure diagram of the library management system [7], as shown in the figure below, this paper takes functional modules as the anchor points, constructs the knowledge graph framework of the Java course that covers the full link of "Basic Knowledge - Practical Application", and clarifies the implementation scenario of each knowledge point in the project.

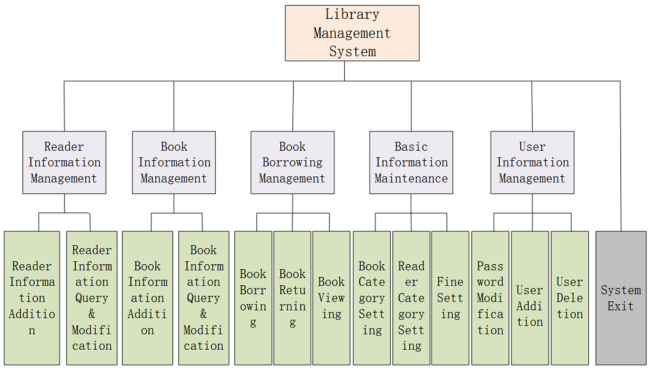



**Fig 2.** Function Module Diagram of Library Management System

#### 2.3.1. Basic Knowledge Layer: The Capability Foundation of the Project

This project covers all the core basic knowledge of the Java entry stage, and each knowledge point has a clear functional implementation scenario:

1. Basic Syntax and Data Types: In the reader information addition and book information addition functions, we use string type to store text information such as reader ID, book ISBN, and name, use integer type to store stock quantity and reader category ID, and use floating-point type to store book price and fine amount, so that students can understand the applicable scenarios of different data types in the process of function implementation.

2. Process Control Structure: In the information entry function, we use conditional judgment (if-else) to complete the validity verification of the input information (such as reader ID format, ISBN rules, and stock range verification); use loop statements to realize the batch entry of reader and

book information, so that students can intuitively understand the practical role of control structures.

3. Object-oriented Basics: We abstract business entities such as readers, books, borrowing records, and system users into Java classes, and use the three major characteristics of encapsulation, inheritance, and polymorphism to construct the entity model of the project, realize the alignment of business logic and code structure, and lay a foundation for the follow-up framework learning.

#### 2.3.2. Practical Application Layer: The Capability Extension of the Project

On the basis of basic knowledge, the project also covers the advanced practical knowledge points of the course:

1. IO Stream and File Operation: It is used to realize the persistent storage of system configuration and borrowing records, and complete the data landing of the basic version.

2. Exception Handling: It is used to handle abnormal scenarios such as insufficient stock, insufficient reader permission, and illegal information entry during borrowing, to improve the robustness of the system.

3. Transaction and Status Management: It is used to ensure the data consistency of services such as book borrowing and returning, and avoid the problem of out-of-sync between stock and borrowing records.

Through this framework, we break the isolated state of knowledge points, so that students can clarify "what problems can knowledge points solve" at the beginning of learning, and avoid abstract learning divorced from practice. At the same time, the project is equipped with specific function implementation examples. In addition to theoretical explanations, we provide sample codes for core logic such as reader information verification and borrowing fine calculation, which is convenient for students to quickly transform theoretical knowledge into practical operations, master Java programming skills in the process of solving practical problems, and better understand the application scenarios of the Java language.

In addition, this model also introduces innovative teaching methods and tools, integrates design thinking to guide students to think about function design from different user perspectives such as administrators and readers, introduces the short iteration concept of agile development, disassembles the project into small tasks that can be quickly implemented, and cultivates students' innovative thinking and engineering practice ability.

## 2.4. Project Introduction: Preparing for Comprehensive Project Development

The core of this model is not the simple superposition of the three types of methods, but takes the library management system project as the anchor point to realize the in-depth integration of the three. The integration of the three runs through the entire teaching process, and the architecture after the collaboration of the three methods is shown in the figure below.

#### 2.4.1. Knowledge Graph + Flipped Preview, Completing Prepositional Foundation

1. Taking knowledge graph[8] as the cognitive skeleton, we solve the problem of "isolated knowledge points", build a complete cognitive link from basic grammar to project practice for students, and let students clarify the application scenario of each knowledge point.

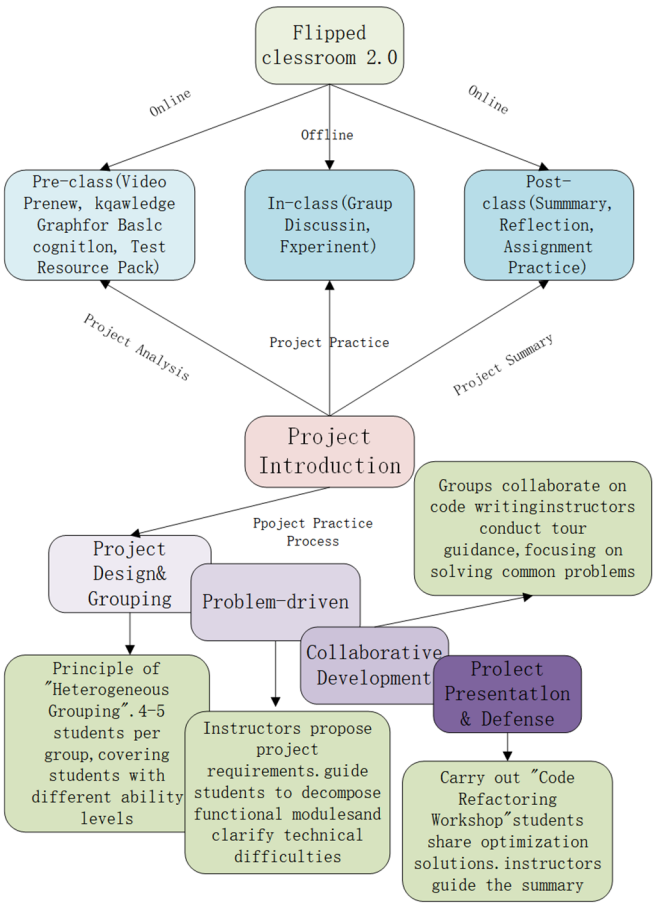



**Fig 3.** Structure Diagram of Teaching Reform Model

2. Taking Flipped Classroom 2.0 as the process carrier, we solve the problem of "disconnection between teaching and learning", move the transmission link of basic knowledge points to the front, release classroom time for in-depth practice and discussion, and highlight the dominant position of students.

3. Taking Project-Based Learning as the ability anchor point, we solve the problem of "disconnection between theory and practice", so that all knowledge learning is carried out around real project requirements, and avoid abstract learning divorced from practice. The three work together to support the full-process teaching activities, and realize the teaching efficiency improvement of 1+1+1>3.

In this stage, we integrate the guiding role of knowledge graph and the task preposition mechanism of flipped classroom: before class, we release preview tasks through Chaoxing Learning Pass, and push the function structure diagram of the library management system, Java technology knowledge graph, and project requirement documents to students synchronously. Students can rely on the knowledge graph to establish the correlation cognition of "knowledge point - function - project" in advance, independently analyze the functional requirements of the five core modules of the project (Reader Information Management, Book Information Management, Book Borrowing Management, Basic Information Maintenance, User Management), sort out the course knowledge points that may be used for each function, and at the same time mark their own knowledge questions, enter the classroom with clear questions, so that the preview link is no longer isolated knowledge point memorization, but advance cognition around the project, to prepare for the follow-up in-depth practice.

#### 2.4.2. Implementation of the Full-process Integrated Teaching

(1) In-class: Flipped Interaction + Project Practice, Realizing In-depth Integration.

In the in-class stage, the in-depth interaction of flipped classroom and the grouped practice of project introduction are deeply integrated. The classroom no longer carries out repeated teaching of basic knowledge points, but focuses on the implementation practice of the project: First, we conduct collective Q&A for the common questions marked by students before class, to clear the knowledge obstacles of preview; Then, instructors put forward the complete project requirements of the library management system, guide students to decompose functional modules combined with the knowledge graph in groups, and jointly analyze the technical difficulties of the project (such as file reading and writing to realize data persistence, and exception handling to ensure system robustness). Then, according to the principle of "heterogeneous grouping", we match students with different foundations into project groups, to ensure that each group has sufficient ability to cover all functions of the project. The groups carry out division of labor and collaboration, and are responsible for the function analysis and preliminary implementation of different modules respectively. Students combine the knowledge points learned before class with the correlation guidance of the knowledge graph, implement them into the function code of the project, complete the transformation from theory to practice, and exercise the ability of team collaboration and division of work at the same time.

(2) Post-class: Project Iteration + Knowledge Complementation, Completing Ability Internalization.

In the post-class stage, the iterative optimization of the project and the supplement and improvement of the knowledge graph are promoted synchronously, to realize the in-depth internalization of knowledge and ability: We establish a multi-dimensional reflection and improvement mechanism, to guide students to complete the project review and iteration: We require students to write project development logs, record the technical difficulties encountered in the project development process, the solution process, and the insights of team collaboration; We require each group to submit code review reports and the improved version of the project code, to optimize the basic version implemented in the class; At the same time, we guide students to supplement and improve their personal knowledge graphs, mark the correlation of the knowledge points used in the project, and complete the internalization and absorption of knowledge; Finally, we assign extended requirements (such as batch import of book information, borrowing renewal function), to guide students to complete the extended development of the project, and realize the continuous improvement of ability.

#### 2.4.3. The Teaching Application Value of the Integrated Model

This three-in-one integrated model has shown significant application value in the teaching practice of Java courses:

1. Accurately solve the pain points of traditional teaching: It completely solves the problems in traditional teaching that students "learn knowledge points alone but cannot implement them into applications" and "cannot connect scattered knowledge points". Through the integration of the three, students can establish a complete cognitive link from the beginning of learning, and realize the in-depth alignment of theoretical knowledge and project practice.

2. Adapt to differentiated learning needs: The pre-class preview of flipped classroom allows students to adjust their learning rhythm according to their own foundation, and the heterogeneous grouping model allows students with different foundations to find a suitable position in the project: students with weak foundations can be responsible for the function implementation of basic syntax, while students with good foundations can be responsible for advanced architecture design and high concurrency scheme discussion, which takes into account the learning needs of students at different levels.

3. Realize the comprehensive cultivation of comprehensive ability: In addition to the improvement of Java programming skills, this model also cultivates students' problem analysis ability, team collaboration ability, and iterative innovation ability, helping students complete the ability transition from "code writers who passively accept knowledge" to "system builders who actively solve problems", which adapts to the ability training requirements of software talents in the new era.

## 3. Expected Effects of the Reform

The expected effects of this teaching reform will be implemented synchronously from two dimensions: students' ability growth and teachers' teaching upgrading.

### 3.1. For Students: Hierarchical Advancement of Ability

Systematic knowledge cognition to deepen knowledge understanding: Relying on the structured display of knowledge graph, it helps students break the dilemma of fragmented learning, establish systematic cognition of Java knowledge points, quickly understand the practical application value of knowledge points combined with the project scenarios of the library management system, and at the same time, rely on the dynamic feedback mechanism to solve knowledge blind spots in a targeted manner, improving learning efficiency.

Project-based practice implementation to improve development ability: Taking the library management system as the real project carrier, we split the project into chapter learning. Students complete module development in groups, and quickly master development and collaboration ability in the practice cycle of flipped classroom. After the course, more than 80% of the students can independently complete the design and implementation of a simple management system.

Problem-oriented learning to strengthen problem-solving ability: Guided by the real problems in the project, combined with the knowledge point correlation of the knowledge graph, we guide students to integrate scattered knowledge points into solutions, exercise their problem analysis and scheme design ability, and realize the transfer and application of knowledge.

Grouped collaborative development to exercise team collaboration ability: Through project grouped development, we exercise students' ability of division of labor, collaboration, communication and exchange, laying the foundation for core soft skills for their subsequent career development.

### 3.2. For Teachers: Strategic Upgrading of Teaching

Strategic reconstruction of teachers' roles: The role of teachers has changed from the traditional "knowledge imparter" to "cognitive architect" and "practice guide": they are no longer the main teaching body in the classroom, but

responsible for building knowledge graphs, designing the teaching framework of project introduction, guiding students' discussion and practice in the classroom, solving students' personalized problems in a targeted manner, and completing the transformation from "teaching knowledge" to "building platforms and guiding directions".

Precision teaching to improve teaching efficiency: Relying on the preview data of the knowledge graph and the questions marked by students before class, teachers can accurately identify students' knowledge blind spots, realize targeted hierarchical teaching, avoid the undifferentiated cramming teaching, and at the same time, intuitively grasp the shortcomings of students' practical ability, realizing real teaching students in accordance with their aptitude.

Reusability and iteration of teaching resources: Relying on the knowledge graph, teaching process, and project cases built with the library management system, a reusable and iterable teaching resource library is formed. Teachers can continuously optimize project cases, update the knowledge graph according to students' learning feedback, realize the continuous upgrading of teaching content, and support the long-term quality improvement of the course.

# 4. Summary

This teaching reform of the "Java Programming" course has constructed a three-in-one teaching model that takes the library management system as the practical carrier, and integrates knowledge graph guidance, Flipped Classroom 2.0, and Project-Based Learning. It effectively solves the pain points of scattered knowledge points and the disconnection between theory and practice in traditional teaching. Through the full-process teaching integration, this model realizes the hierarchical advancement of students' knowledge cognition and practical ability, and at the same time forms a reusable and iterable teaching resource system, providing a practical reference for the teaching reform of similar computer basic courses.

# Acknowledgments

This work was supported by the School-level Quality Engineering Project: Practical Research on the Integrated Teaching Reform of Knowledge Graph-enabled Flipped Classroom 2.0 and Project Introduction in the Java Programming Course. (project number: GKZLGC2025049).

# References

[1] Smith, J., Johnson, A., & Williams, B. (2023). Flipped Classroom 2.0: Enhancing programming education with knowledge graphs. *Computers & Education*, *198*, 104789. https: //doi.org/10.1016/j.compedu.2023.104789.

[2] Zhang, W., & Wang, L. (2021). Practice of project-based learning in Java programming course. *Higher Education of Sciences*, (5), 112–116.

[3] Chen, X., & Huang, Q. (2020). Teaching reform of Java course for software engineering major based on PBL. *Software Engineering*, *23*(9), 62–64.

[4] Liu, F., & Zhou, M. (2024). Construction of knowledge system in programming courses driven by knowledge graph. *Computer Education*, (2), 89–93.

[5] Zhao, W., & Sun, L. (2023). Teaching practice of flipped classroom in Java course supported by Chaoxing Learning Pass. *Computer and Information Technology*, *31*(3), 92–94.

[6] Wu, P., & Zheng, X. (2022). Exploration on teaching reform of Java course for engineering education accreditation. *Computer Education*, (7), 98–102.

[7] Lee, S., & Park, J. (2022). Project-based learning with knowledge visualization for Java programming courses. *Journal of Educational Technology & Society*, *25*(2), 145–158.

[8] Garcia, M., & Rodriguez, L. (2021). Knowledge graphs for personalized learning in programming education. *IEEE Transactions on Education*, *65*(4), 378–387. https://doi.org/ 10. 1109/TE.2021.3106324.